

学生プロジェクトが直面した問題事例と アジャイルによる対処可能性の考察

平山 雅樹^{†1} 新野 朝丈^{†1} 児玉 公信^{†2}
松澤 芳昭^{†3} 太田 剛^{†3}

学生が初めてのソフトウェア開発プロジェクトを経験する中で直面した問題について報告する。対象とする開発プロジェクトは、企業が実際のシステムを外注する際に使用した仕様書を基にしており、企業のソフトウェア開発と同程度の品質を目指すものである。プロジェクトの規模は10人月程度で、アジャイル開発プロセスが試みられた。現在進捗している段階まで、生じた問題について記述して種類を整理した結果、作業の目的の理解不足とリスクの意識不足に起因する問題であることが分かった。

A Report of Experiences for the Students' First IS Project using Agile Process

MASAKI HIRAYAMA,^{†1} TOMOTAKE NIINO,^{†1}
KIMINOBU KODAMA,^{†2} YOSHIAKI MATSUZAWA^{†3}
and TSUYOSHI OHTA^{†3}

In this paper, we report our experience that students faced in the first IS (Information Systems) development project. The project was driven by the RFP (Request For Proposal) which was used in the real situation when the project owner ordered to developers. Although the developers were not students, students tried to develop the system as the same quality as professionals do. Agile process was applied to the project, and the size of the project is approximately ten man-months. Now the project has proceeded to the middle of the goal, and problems that were happened in the project until now were described and classified. Then we have found that the problems are caused by “losing the objective of the work” and “lack of focusing the risk management”.

1. はじめに

近年、多くのIT人材育成プログラムが日本各地で提案されている。こうした議論のなかで、産学連携等の社会に開かれた環境で、プロジェクト形式の実践的な形態を取って教育を行うことについては、合意が得られてきたように思われる。

しかしながら、実利用されるソフトウェア開発の指導法に関しては、ほとんど何も合意は得られておらず、担当者の試行錯誤で実施されているのが実状である。この問題を解決するために、2009年度から始まったPBL(Project-Based Learning)教材洗練ワーキンググループ(PBL-WG)では、PBL担当者のノウハウを集めたノウハウ集を編纂しているが、その議論の内容は参加した教育者の経験にのみ基づくものである。

指導法をより豊かにしていくためには、指導法確立の根拠としてのプロジェクト事例を積み上げる必要がある。本稿の目的は、その事例の一つを提供することである。学生が実施したプロジェクトをそのプロセスと共に記述して、生じた問題についての整理を試みる。

本稿のもう一つのキーワードは、「アジャイル」¹⁾である。取り上げるプロジェクトの特徴では、アジャイル開発プロセスの適用を試みている。研究プロジェクトは不確定要素が大きいため、ウォーターフォールと比してアジャイルプロセスが適していると思われる。しかしながら、学生プロジェクトとしても、現場の開発としても、日本では適用報告が少ないのが現状である。それらを蓄積し、アジャイルを教育するためのグッドプラクティスを提供することが本研究の目標である。

2. OJLとは

本章では、プロジェクト説明に先立って、OJL(On the Job Learning)について簡単に述べる。OJLは「先導的ITスペシャリスト育成推進プログラム」の拠点として、名古屋大学他4大学が連携して提案した「OJLによる最先端技術適応能力を持つIT人材育成拠点の形成」プログラムの中核をなす教育システムである²⁾。OJLでは、現実のソフトウェア開発プロジェクトを用いて、大学と企業が一体となって実践教育である。現実の開発プロジェ

^{†1} 静岡大学大学院情報学研究所
Graduate School of Informatics, Shizuoka University

^{†2} 情報システム総研
Information Systems Research Institute

^{†3} 静岡大学情報学部
Faculty of Informatics, Shizuoka University

クトを活用するため、納期や予算といった企業における現実的な制約を考慮してプロジェクトを進める必要がある。産業界の協力のもと、実際のソフトウェア開発を題材にすることで、開発技術の実践的な内容を深く理解させ、それが訓練ではなく教育となるように教員が指導を行うという点が特徴である。

3. OJL プロジェクトの内容

本章では我々が実施している OJL プロジェクトについて述べる。

3.1 OJL プロジェクトの背景

本 OJL プロジェクトのテーマは、「CPD プログラムポータルシステム」（以降、「CPD ポータル」と略す）の開発である。当テーマは、日本工学会が実際に外注したシステムであり、実際のソフトウェアベンダーが受注開発しているものである。本 OJL プロジェクトは、正規の受注企業とは独立したプロジェクトとして、学生が並行して開発を行うものである。

3.2 OJL プロジェクトの組織

本 OJL プロジェクトの名称は Cat-Engine である。この Cat-Engine プロジェクトの協力企業は情報システム総研である。プロダクトオーナーは同社の児玉公信（第 3 著者）、開発メンバは静岡大学大学院の平山雅樹（第 1 著者）、同じく新野朝丈（第 2 著者）である。プロジェクトの規模は 10 人月程度と想定した。なお、開発するシステムの名称も Cat-Engine システムである。

3.3 OJL プロジェクトの内容

このシステムは、Web 上に散在している日本工学会 CPD 協議会に所属する学協会が主催する講演や講習会、セミナーの情報（以下、「CPD プログラム情報」と記述）を収集、集約して、日本工学会のページとして提示する Web システムである。この要求定義文書は日本工学会の Web ページにある構想書³⁾、提案依頼書⁴⁾より参照することができる。

3.4 開発メンバの知識レベル

開発メンバは当該 IT スペシャリストコースのカリキュラムに沿った、ソフトウェア工学の座学⁵⁾、及び規定の課題を用いた PBL(Project-Based Learning) 形式の演習⁶⁾を受講済みである。

3.5 システムの目的

情報処理学会をはじめとした多くの学会（以下、学協会と記述）では、技術者の継続研鑽（CPD ; Continuous Professional Development）のために、多くの講演やセミナーを提供している。これらの開催情報はほとんど Web で提供される。CPD プログラムの受講を検討

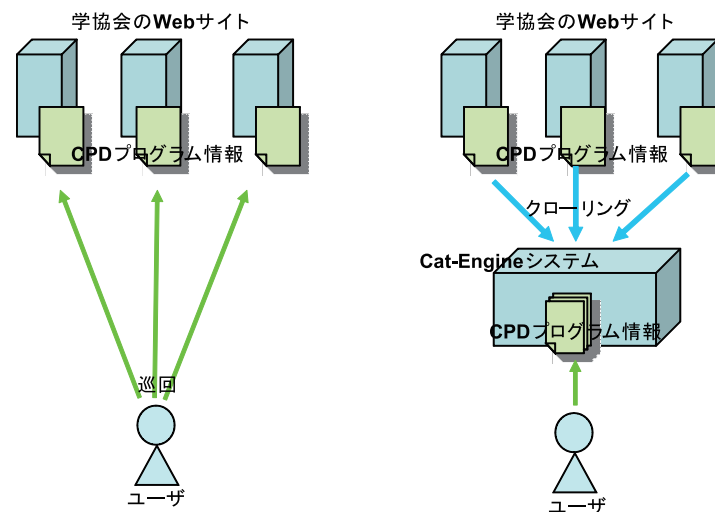


図1 開発システムの概要（左側が従来のシステム、右側が本システム）

しているユーザは、複数の学協会の CPD プログラム情報を横断的にチェックするために複数の学協会のサイトを巡回しなければならないという問題がある。Cat-Engine システムの目的は、複数の学協会のサイトに分散している CPD プログラム情報をクロールによって収集、集約して提示することによって、受講を検討している技術者に巡回の手間を減らすと同時に、受講の機会を見逃さないよう支援することである。CPD プログラムは学協会のサイトごとに独自の提供形式となっている。この様子を図 1 に示す。

3.6 Cat-Engine システムの機能

Cat-Engine システムは、プロジェクトの目的を達成するために以下の機能を提供する。

- CPD プログラム情報収集機能
 - 学協会の事務員が CPD プログラム情報を入力する機能
 - サイトごとの CPD プログラム情報の記述方法を記したファイル（以下、テンプレートと記述）を生成する機能
 - クロールングによって学協会のサイトを巡回し、情報を登録する機能
- CPD プログラム情報提示機能
 - CPD プログラム情報を検索する機能

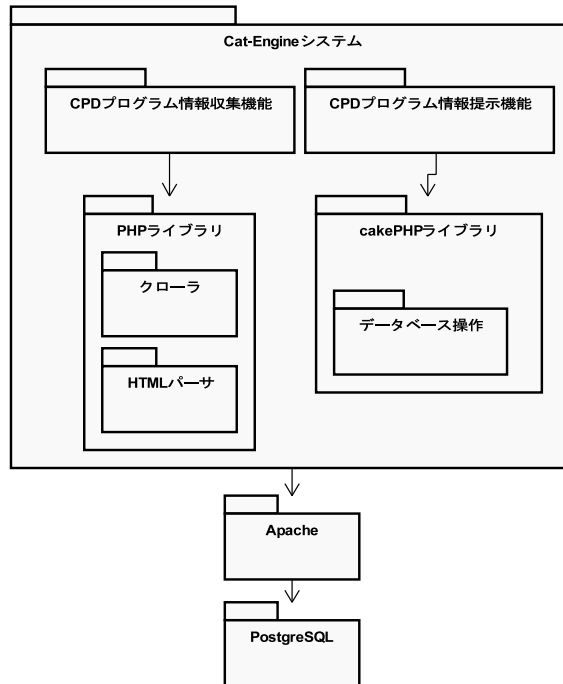


図 2 Cat-Engine システムの構成

– CPD プログラム情報をソートする機能

3.7 ソフトウェアアーキテクチャ

Cat-Engine システムの開発に使用する言語は PHP、サーバソフトウェアは Apache、PHP フレームワークは cakePHP、データベースは PostgreSQL を用いる。ソフトウェアの構成を図 2 に示す。本プロジェクトでは、図 2 で示すパッケージの内、CPD プログラム情報収集機能と CPD プログラム情報提示機能の実装を行う。

3.8 データベーススキーマ

データベーススキーマは予め構想書³⁾ に例示されているため、初期モデルとしてそれを用いている。構想書³⁾ に記載されているスキーマを図 3 に転載する。

3.9 開発プロセス

本プロジェクトは、アジャイル開発プロセスの採用を試みている。アジャイルプロセスの

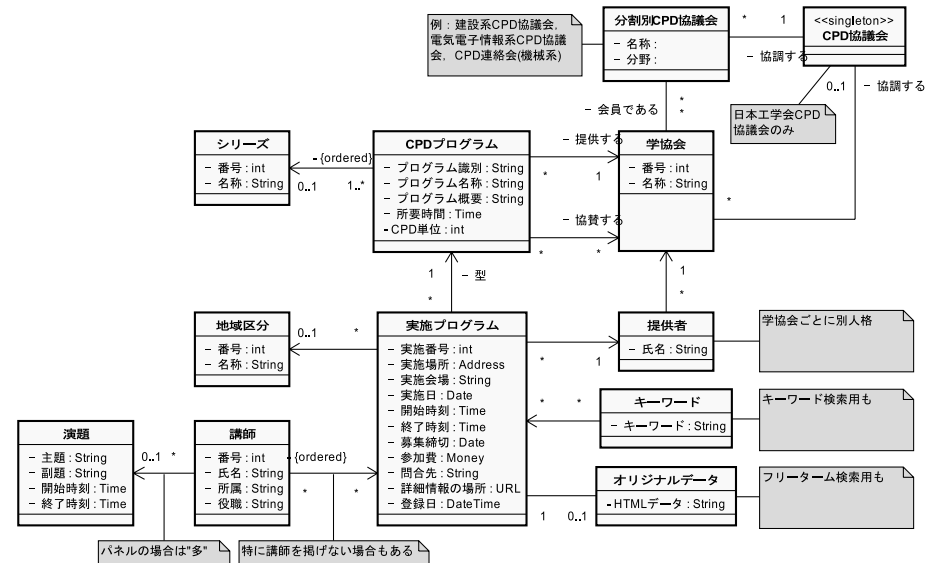


図 3 データベーススキーマ (構想書³⁾ より転載)

採用は提案依頼書⁴⁾ での要求事項である。

実際の開発計画について、まず当初計画のガントチャートを図 4 に、修正計画のガントチャートを図 5 に示す。当初計画は、開発初期に学生が立案した第 1 バージョンの計画である。修正計画は、プロダクトオーナー及び指導教員によって当初計画のレビューを実施した修正後の計画である。

当初計画は全体的に見て、ウォーターフォール型のプロセスに非常に似ている傾向にある。なぜなら、一つの機能を完了してから次の機能を実装するという計画になっており、設計期間と実装期間とはっきりと分かれているからである。当初ウォーターフォールでの作成を予定していたわけではなく、アジャイルのつもりで作成した予定である。

一方で、修正計画はスパイラル型のプロセスとなっている。アジャイル開発プロセスとして、より適した形になったといえよう。同じ機能を何度もリリースすることによって、要求の変更への対応が柔軟になっている。

当初計画と修正計画では実装する機能の優先順位が異なっている。すなわち、当初計画ではユーザインタフェースを優先的に実装する計画となっていたのに対し、修正計画では

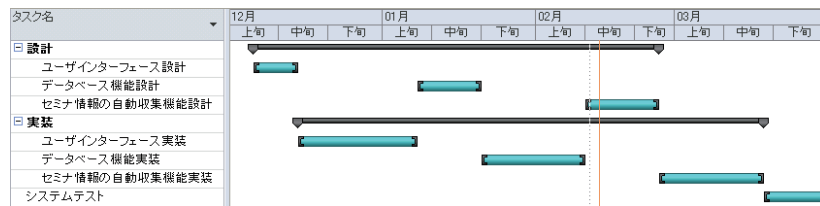


図 4 当初計画

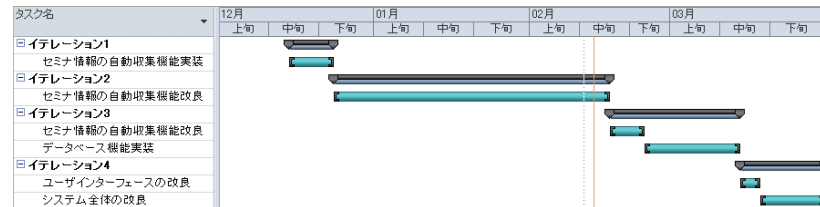


図 5 修正計画

CPD プログラム情報の収集に関する機能を優先的に実装する計画となっている。

3.10 現在の進捗

システム提案, 初期設計, 第 1 回目イテレーションは既に完了している。現在は, 第 2 回目イテレーションの実施中である。

4. 直面した問題

本章では, プロジェクトを進めていく中で, 我々が直面した問題について, 時系列順に列挙する。各問題に対しては, 名前, 工程, 内容, 学生の言い分を挙げる。

問題 1: 提案書レビューにて, コストと運用に関する記述不足を指摘された

工程 システム提案

内容 システム提案書において, システムの保障とランニングコストに関する記述の具体性がなかった。

学生の言い分 たくさんの参考資料からどれを選べばよいのかが分からなかった。

問題 2: 提案書レビューにて, 「お客様に読んでもらうという意識がない」と指摘された

工程 システム提案

内容 システム提案書について, お客様が知りたい情報 (性能, 価格, システムのウリなど) を押し出していなかった。スケジュールについてガントチャートの形になっておらず, 読みやすい形になっていなかった。

学生の言い分 読み手がさっと読めるような文書にしようという考えがなく, 提案依頼書で聞かれた事について答えるということしか意識していなかった。

問題 3: 見積調査で時間を浪費した

工程 システム提案

内容 システム提案書を作るにあたり, 性能要求について考慮しなかったため, ハードウェアやソフトウェアに対して, 何を見積もればいいのか分からず, 調査の基準がぶれて時間を浪費した。

学生の言い分 サーバプログラミングの経験がなく, またサーバを扱うことも始めてであったため, 何が必要となるのか分からなかった。

問題 4: 提示価格が乱高下した

工程 システム提案

内容 ソフトウェア開発費について, 最初は市場価格を調査し一番安いところに合わせた金額として, 122 万円 (内ソフトウェア開発費 40 万円) を提示した。その際に, 赤字になってしまうという指摘を受けたため, 人月による見積もりを行い, 提示価格を 380 万円 (375 万円) に変更した。今度は高すぎるという指摘を受けたため, 最終的には, 人月に学生補正をかけることで, 市場価格とのバランスを取り, 100 万 (90 万) に変更した。

学生の言い分 お客様がそのシステムにどれくらいまでお金をかけてくれるのかを考えていなかった。また, 市場価格と人月コストとお客様の要望のバランスを取るということを考えていなかった。

問題 5: システム提案書の提出期限が守られなかった

工程 システム提案

内容 提出期限を 1 日超過した。

学生の言い分 開発システムの説明および各工程における作業内容の説明を書くのに時間がかかり, さらに開発メンバの一人がインターンシップの都合で提出期限直前に 1 日作

業が出来なかった。一つ一つの項目について、質を時間に合わせるという意識がなく、途中で見切りを付けるということをしなかった。

問題 6: 提案書レビューにて、「利用者が持つスキルへの配慮不足」を指摘された

工程 システム提案

内容 テキストベースの言語を一般事務員が扱う仕様になっていた。一般事務員すべてに、テキストファイルを直接編集してテンプレートを作成する能力を期待することは出来ないと指摘された。

学生の言い分 自分達の「こういうことができたらいいのではないか」という観点だけで仕様を決めてしまい、ある程度の知識やスキルがあることを前提として考えていた。

問題 7: 設計レビューにて、ユースケースの書き方の不備を指摘された

工程 初期設計

内容 ユースケース記述の目的に対して、ユースケース名と同じものが書かれていた。また、シナリオが省略されていた。

学生の言い分 ユースケース記述を間違っ解釈しており、ユースケース名と目的がほぼ同じものになっていても、おかしいと感じなかった。また、シナリオを単なる具体例と考えてしまい、省略していた。

問題 8: 設計レビューにて、「細かい部分まで定義しすぎている」と指摘された

工程 初期設計

内容 ユースケース記述の例外処理として、空入力への対処といった細かい部分まで定義を行った。現時点でそこまでやる必要はないという指摘を受けた。

学生の言い分 設計の段階で細かい部分まで100%定義すれば、良いシステムが出来ると考えていた。

問題 9: 定例ミーティングにて、「優先度の判断が間違っている」と指摘された

工程 第1回目イテレーション

内容 初期スケジュールでは、UIやデータベースを操作する機能を優先して実装するスケジュールとなっていた。技術的なリスクを解消することを優先すべきだという指摘があり、リスクの解消を優先したスケジュールに変更された。

学生の言い分 お客様はUIから作ることを要望していると考えていたが、実際には技術的なリスクを潰すことが優先されていた。

問題 10: 同じ機能を持つモジュールが2重に作成された

工程 第1回目イテレーション

内容 開発メンバが別々にHTMLファイルを読み込むためのモジュールを作成し、同じ機能を持つモジュールが2重に作成された。

学生の言い分 簡単なモジュールであったため、相談をせずに自分で勝手に作ってしまった。

問題 11: 定例ミーティングにて、「アーキテクチャが決まらないため、第2回目イテレーションは終了できない」と指摘された

工程 第2回目イテレーション

内容 「技術的なリスクを解消して、アーキテクチャを決定できなければ、第2回目イテレーションは終了できない」と指摘されたため、第2回目イテレーションの期間を延長した。

学生の言い分 計画を立てる際に、何がリスクの高い問題になるのかということ深く考えていなかった。

問題 12: プロダクトオーナーの一言により仕様が揺れ動いた

工程 第2回目イテレーション

内容 「できるだけ利用者の手間が少なくなる方が良い」というプロダクトオーナーの言葉を受け、仕様変更にかかる技術的コストの調査をせずに、仕様を変更した。その後、変更した仕様では品質を維持できないことが判明し、仕様を元のものに再変更した。

学生の言い分 お客様の要望全てに答えようとしていた。

問題 13: 細かいことに対して、長時間の議論が発生した

工程 全工程

内容 ささいなことに対する長時間の議論が、週1回程度の頻度で発生した。例えば、データベースの識別子について、 세미나名称と実施日の他に、セミナー主催学協会名とセミナー掲載学協会名のどちらを使用するかということで議論になった。その場では決着がつかなかったが、後日、プロジェクトマネージャの方から構想書において指定されていると

いう指摘を受け、主催学協会名を使用することとした。

学生の言い分 お互いの意見を主張する際に、自分の意見の利点をちゃんと説明することが出来ていない。

5. 考 察

第4章で列挙した各問題について考察、分類を行った。本章ではその結果を述べる。

5.1 問題の分類軸

4章で列挙した各問題について、以下の3つの分類軸を考えた。

(A) 目的を理解しないで作業をしてしまったために起きた問題

その作業の目的が何なのか明確でないまま、作業を進めてしまったために発生した問題を表している。例えば、4章の問題1では提案書をどのような目的で書くのかを明確にせず提案書を記述したため、コストと運用に関する記述に関する不備を指摘された。

(B) スケジュールマネジメント、リスクマネジメントに不備があったために起きた問題

プロジェクトを進めるなかで、作業をすすめることに夢中になって、リスクやスケジュールのマネジメントに時間を掛けなかったために発生した問題を表している。例えば、問題9ではどの機能のリスクが高いかを考えていなかったために、優先順位の修正が発生してしまった。

(C) アジャイル開発について理解が不足していたために起きた問題

これは、アジャイルだと思って実施したプロセスがアジャイルではなかったために発生した問題を表している。例えば、問題8ではウォーターフォール型の設計を行ってしまい、過度に詳細な設計を行ってしまった。

5.2 問題の分類

5.1節にて述べた分類軸を利用して、4章で列挙した問題を分類した。その結果を表1に示す。列A,B,Cは対応する分類軸を表現しており、該当するものには○、ある程度は該当するものには△を記している。

5.3 考 察

問題の発生時期と、問題数の関係を図6にまとめる。このとき、問題数は表1の○を1、△を0.5として換算した。

図6より、システム提案や初期設計といったプロジェクト前半においては、分類軸Aについて指摘されることが多かったことが分かる。プロジェクト中盤になると、分類軸Bの発生

が増えている。分類軸Cについては、プロジェクト前半から中盤にかけての各工程で、まっぴらなく問題が発生している。

分類軸Aの問題の原因は、参考資料の手順の通りに実施すればよいという考えにあったと考える。参考となる資料を探して、その形式を模倣し、とにかく必要事項を埋めることに集中してしまった。そのため、お客様が求めている事項に時間をかける、お客様が必要としている事項について詳しく記述していないという問題が発生した。誰がどんな意図を持って読む文書であって、どのような項目がどのくらいのレベルで記述できれば十分であるかの基準をしっかりと決める必要があったと考える。

分類軸Bの問題に関しては、一つの作業を納得いくまで続けようとしたことに原因があると考えられる。作業の終了目標を具体的に設定していなかったため、いつまで経っても作業を終了させることができず、結果としてスケジュールの遅延が発生する、よりリスクの高いタスクを見落とすという問題が発生した。この失敗を通じて、終了目標を明確に設定しなければ、どんなに時間を掛けても作業が終わらないということが分かった。

分類軸Cの問題の原因としては、我々がウォーターフォール型開発しか学習経験がなく、その経験に引きずられたことが考えられる。アジャイルの手法を学ぶにしても、専門書を1か

表1 問題の分類

| 工程 | 問題内容 | A | B | C |
|-------------|--|---|---|---|
| システム提案 | 1 提案書レビューにて、コストと運用に関する記述不足を指摘された | ○ | | |
| | 2 提案書レビューにて、「お客様に読んでもらうという意識がない」と指摘された | ○ | | |
| | 3 見積調査で時間を浪費した | ○ | | △ |
| | 4 提示価格が乱高下した | ○ | | |
| | 5 システム提案書の提出期限が守られなかった | | ○ | |
| | 6 提案書レビューにて、「利用者が持つスキルへの配慮不足」を指摘された | ○ | | △ |
| 初期設計 | 7 設計レビューにて、ユースケースの書き方の不備を指摘された | ○ | | |
| | 8 設計レビューにて、「細かい部分まで定義しすぎている」と指摘された | ○ | | ○ |
| 第1回目イテレーション | 9 定例ミーティングにて、「優先度の判断が間違っている」と指摘された | | ○ | △ |
| | 10 同じ機能を持つモジュールが2重に作成された | | ○ | |
| 第2回目イテレーション | 11 定例ミーティングにて、「アーキテクチャが決まらないため、第2回目イテレーションは終了できない」と指摘された | | ○ | △ |
| | 12 プロダクトオーナーの一言により仕様が揺れ動いた | | ○ | ○ |
| 全行程 | 13 細かいことに対して、長時間の議論が発生した | ○ | | |

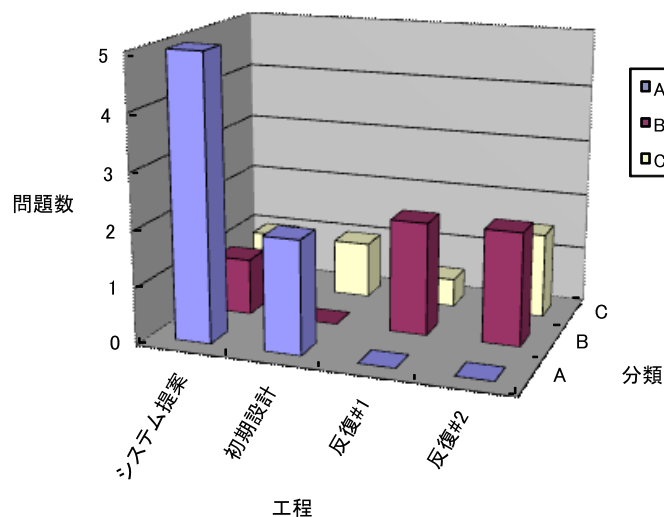


図 6 工程別の問題発生数

ら読まずに必要と感じる部分だけを読んだため、そこで得た断片的な知識とウォータフォール手法が組み合わされてしまったと考えられる。

5.4 今後の展望

さらにプロジェクトを進めていく中で、より多くの問題についてのデータを収集し、学生が初めてプロジェクトを実施する際の流れのモデル化を行う事が今後の課題である。さらには、実プロジェクト上で教育を行う際に、できるだけ避けるべき問題とできるだけ体験させるべき問題を分類し、教育者のための資料として公開することが有用ではないかと考えている。

6. まとめ

本稿では静岡大学で実施中の産学連携によりソフトウェア開発を実施する OJL(On the Job Learning) プロジェクトの一事例を報告した。

当該 OJL プロジェクトのテーマは、実際のソフトウェアベンダーが受注開発している、

「CPD プログラムポータルシステム」であり、本 OJL プロジェクトは、正規の受注企業とは独立したプロジェクトとして、学生が並行して開発を行うものである。

開発システムはソフトウェア工学を一通り学習した大学院生が初めて挑戦する 10 人月規模のシステムであり、アジャイルプロセスが採用された。4 回中 2 回目の反復プロセスがほぼ終了した段階までに起きた 13 件の問題について述べた。それらの問題を 3 つの分類軸によって分類・考察した結果、「評価基準を決めた上で作業を行う」、「納期に間に合う終了目標を明確にする」「目的意識を持って技術適用を行う」ことが学生プロジェクトの初期の課題となることが分かった。

今後も継続してデータを収集し、学生が初めてプロジェクトを実施する際の問題発見解決プロセスのモデル化を行う事が今後の課題である。

謝辞 本プロジェクトの全工程において、親切丁寧なご指導を頂きました静岡大学情報学部情報科学科の塩見彰睦教授、秋川友宏准教授、野口靖浩特任助教に厚く御礼申し上げます。

参考文献

- 1) James Shore and Shane Warden: アート・オブ・アジャイル開発-組織を成功に導くエクストリームプログラミング-, オライリージャパン (2009).
- 2) 小林隆志, 沢田篤史, 山本晋一郎, 野呂昌満, 阿草清滋: On the Job Learning: 産学連携による新しいソフトウェア工学教育手法, 情報システム学会誌, Vol.5, No.2, pp. 44-56 (2010).
- 3) 日本工学会 CPD 協議会: 日本工学会 CPD プログラムポータル構想書 http://www.jfes.or.jp/topic/topic20100823-cpd-portal_offer_plan.pdf (2011/02/12 に参照).
- 4) 日本工学会 CPD 協議会: 日本工学会 CPD プログラムポータル開発提案依頼書 http://www.jfes.or.jp/_cpd/topic/topic20100823-cpd-portal_offer_rfp.pdf (2011/02/12 に参照).
- 5) 沢田篤史, 小林隆志, 金子伸幸, 中道上, 大久保弘崇, 山本晋一郎: 飛行船制御を題材としたプロジェクト型ソフトウェア開発実習, 情報処理学会論文誌, Vol.50, No.11, pp.2677-2689 (2009).
- 6) 松澤芳昭, 塩見彰睦, 秋川友宏, 酒井三四郎: ソフトウェア開発の教員主導型 PBL における反復プロセスと EVM 導入の効果, 情報処理学会研究報告 (CE-99-9), pp.1-8 (2009).